Remarks

Response to Restriction Requirement:

Responsive to the Restriction Requirement stated in the Office Action, applicants have elected without traverse Group A, claims 1-6 and 29-34.

The Drawings:

The Drawings have been amended to correct the margins of sheets 3, 4, 7, 9-11 and 13. FIG. 4C has been amended to remove a duplicate numeral 70. FIG. 8 has been amended to replace the word multiplexor with the word multiplexer. A letter entitled Corrections to the Drawings that explains these changes is enclosed, along with replacement drawings for sheets 3, 4, 6, 7, 9-11 and 13.

The Specification:

The specification has been amended to correspond to the drawings.

The Claims:

The claims have been amended to comply with the restriction requirement and to clarify the invention. Several new claims have been added that are also believed to comport with the restriction requirement and to be patentable over the cited references.

Claims 1-6 and 29-34 stand rejected under 35 U.S.C. § 103 as being obvious over Bilansky et al., U.S. Patent Number 5,878,225 (hereinafter Bilansky), in view of Radogna et al., U.S. Patent Number 5,878,225 (hereinafter Radogna).

Independent Claim 1:

Regarding independent claim 1, the Office Action states:

> Bilansky discloses a method for bypassing protocol layers (see abstract). Bilansky discloses the invention substantially as claimed. Taking claim 1 as an exemplary claim, Bilansky discloses a method for communication between a network and a host computer having a processor and a sequential stack of protocol layers (see figures 1-4, client 100, server 200 and protocol layers 110,130,150, 170, 190), wherein the method comprising;

receiving, by said host from said network, a message packet including data and a plurality of headers corresponding to said stack of protocol layers, said data intended for placement in a destination of said host according to protocol processing of said headers, processing said plurality of headers, including creating a group of headers, and choosing whether to process said packet by said protocol layer (see abstract, column 2, lines 25-50, column 3, line 65 to column 4 line 6, column 4, lines 40-67, column 6, lines 5-45, DDM fast-path bypasses the communication service layers).

Bilansky does not explicitly show the process of processing plurality of headers as a group and creating a summary of group of headers. However, Bilansky does show the processing of headers (see column 5, line 24 to column 6, line 66). Radogna discloses a method for processing messages in a communications network similar to that of Bilansky, wherein Radogna discloses the process of processing plurality of headers as a group and creating a summary of group of headers (see column 3, line 12 to column 4, line 50, RHP 46 for parsing headers). Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Bilansky in view of Radogna by including the process of processing plurality of headers as a group and creating a summary of group of headers, because Bilansky suggests the processing of headers. One of ordinary skill in the art would have been motivated to modify and use the method disclosed by Bilansky since Bilansky suggests that bypassing the protocol stack can reduce the communication overhead (see column 10, lines 25-35).

In the invention of Bilansky, a received message packet does not include both data and headers corresponding to a stack of protocol layers. Instead, it appears that Bilansky transfers packets of data information on a data path and packets of control information on a control path (Abstract, lines 11-14; Summary, column 2, lines 35-38; Best Mode, column 4, lines 27-31). Thus Bilansky appears to establish a protected fast-path over a network connecting a client or source and a server or target (see, e.g., column 4, lines 21-38). Applicant's invention of amended claim 1 on the other hand recites "receiving ...a message packet including data and a plurality of headers" and then "choosing ... whether to process said packet by said protocol layers or to avoid processing by said protocol layers."

Since the invention defined by claim 1 is applicable to many common message packets, such as data packets having TCP/IP headers, it is much more adaptable than the protected data packets of Bilansky, which appear to require establishing a protected fast

C

path on a dedicated network. Stated differently, the data packets of Bilansky evidently would not be transferred over networks that comply with different protocols than the "Advanced Program-to-Program Communication (APPC)" protocol described in Bilansky. The data packets of Bilansky would evidently also not be transferred over an internetwork containing non-APPC networks, such as the Internet, further restricting Bilansky's disclosure compared to the invention defined by amended claim 1.

Moreover, applicants submit that one of ordinary skill in the art would not have been motivated to add headers to the data packets of Bilanksy, because having separate data and control packets appears essential to the separate data and control paths that allow data to travel over the fast path of that invention. In other words, having control information attached to data on Bilansky's data path would delay if not destroy Bilansky's fast path, negating the main purpose of Bilansky's invention.

Also note that, unlike amended claim 1, the data of Bilansky is not "intended for placement in the host according to protocol processing of the headers," but instead appears to be directed according to the network fast-path that has been set up with the separate control packets.

Further, Bilansky does not teach "processing, sequentially as a group, the plurality of headers," unlike amended claim 1. As noted above, headers for packets containing data are not employed with Bilansky's invention, and are not processed as defined in amended claim 1. The details of processing control packets are not discussed in Bilansky, implying that such processing is performed conventionally with a CPU running a protocol stack that individually processes each protocol layer of control information, as opposed to processing sequentially as a group.

Also, as noted in the Office Action, Bilansky does not teach "creating a summary" of the group of headers, unlike amended claim 1. Such a summary would not benefit the separate control packets and data packets of Bilansky, and thus modifying Bilansky to involve such a summary would at best consume scarce resources for meaningless tasks, and at worst would destroy the functioning of that invention.

In further contrast to amended claim 1, Bilansky does not teach "choosing, dependent upon said summary, whether to process said packet by said protocol layers or to avoid processing by said protocol layers." Instead, Bilansky determines whether or not

to establish a protected path before data packets are transmitted between the server and client of that invention (see, e.g., column 4, lines 21-38). Thus the fast or slow network path of Bilansky is selected before the data packets or control packets are sent, avoiding layers of the path "when performing certain preselected functions" (column 3, lines 59-60).

Radogna, on the other hand, teaches translating data link layer and network layer frame headers for a switch, bridge or router that forwards entire network frames. Radogna does not teach or suggest a way of performing the more complex task of placing the data from the frames in the correct "destination" of a receiving host, in contrast to amended claim 1. Thus Radogna does not teach "receiving, by said host from said network, a message packet including data and a plurality of headers corresponding to the stack of protocol layers," where the data is "intended for placement in said host according to protocol processing of said headers." Radogna instead teaches forwarding a whole network frame, including data and headers, with optional translation of data link or network layer headers.

Radogna also does not teach "choosing, dependent upon said summary, whether to process said packet by said protocol layers or to avoid processing by said protocol layers, for storing said data in a destination in said host," in contrast to amended claim 1. As mentioned above, Radogna teaches forwarding a whole frame including data and headers, and does not provide the possibility of landing the data in a destination in the host, free of headers.

Moreover, applicants submit that one of ordinary skill in the art also would not have attempted to modify Bilansky by employing the header processing of Radogna, as Bilansky would have been expected to work less well as modified. For instance, the data packets of Bilansky would not benefit from the header translation of Radogna, since those data packets are free of headers. Further, adding headers to those data packets is not only unnecessary for the fast-path of Bilansky, but would appear to subvert that fast-path, since the headers would then be interspersed with the data at the receiving client or server of Bilansky, requiring processing or removal of the headers, in either case slowing Bilanksy's fast-path. Similarly, processing the data link and network layer headers as taught by Radogna would not be useful for Bilansky, because Bilansky does not include

C

such headers with data packets and does not need routing over its protected fast-path between client and server. Likewise, creating a summary of the group of headers would also not be useful for Bilansky, because such a summary is not needed for and would likely slow down Bilanksy's fast-path.

Finally, assuming for the sake of argument that one of ordinary skill in the art would have modified Bilansky with Radogna, the resulting combination would have been substantially different than that defined in the claims at issue. For instance, amended claim 1 defines, in part, "choosing, dependent upon said summary, whether to process said packet by said protocol layers." Bilansky makes the choice of a fast-path or slow-path based on preselected functions, such as sending and receiving data, independent of any summary. Thus, even if header layers were added to the data packets of Bilansky in spite of the disincentives mentioned above, Bilansky would still not make a choice of paths "dependent on said summary" of a received packet, because Bilansky's path has been preselected.

For at least the foregoing reasons, applicants respectfully submit that amended claim 1 is nonobvious over Bilansky in view of Radogna.

Dependent Claims 2-6:

Regarding dependent claim 2, the Office Action states:

> As per claim 2, Bilansky teaches the method of claim 1 wherein said method further comprises the steps of sending said data to said destination according to said summary of said group without processing said headers by said protocol layers (see column 6, lines 18-45, bypassing communication service layers).

Dependent claim 2 has been amended to define the method of claim 1, further comprising transferring said data without said headers to said destination in accordance with said summary of said group, without processing said headers by said protocol layers. In addition to all the reasons why amended claim 1 is nonobvious over Bilansky in view of Radogna, amended claim 2 defines transferring the data without the headers to the destination in the host, in accordance with the summary of the group. As noted above, Radogna translates headers for forwarding packets containing both data and headers, routing the packets along or between networks, and never suggesting splitting the data

from the headers. Bilansky, in contrast, selects a protected path prior to sending data packets that contain no headers. One of ordinary skill in the art would not have modified the invention of Bilansky by adding headers, and then processed the headers sequentially as a group to create a summary, and then transferred the data without the headers to the destination in the host, in accordance with the summary of the group, as all of this would be have been expected to slow down the fast-path of Bilansky.

Regarding dependent claim 3, the Office Action states:

> As per claim 3, Bilansky teaches the method of above claims, wherein said method further comprises the steps of wherein said processing of said group of headers occurs during said receiving, by said host from said network, of said message packet.

As noted above, Bilansky does not receive at a client or server a message packet containing both data and headers, and thus does not process the headers of such a packet as defined in dependent claim 3.

Regarding dependent claim 4, the Office Action states:

> As per claim 4, Bilansky teaches the method of above claims, wherein said method further comprises creating a communication control block for a connection including said packet, and matching said communication control block, for sending said data to said destination (see column 5, line 5 to column 6, line 61).

As noted above, Bilansky does not receive at a client or server a message packet containing both data and headers, and thus does not process the headers of such a packet to create a summary of the packet. In contrast with amended claim 4, Bilansky also does not create a communication control block to which the summary is matched for transferring the data to the destination.

Regarding dependent claim 5, the Office Action states:

> As per claim 5, Bilansky teaches the method of above claims, wherein method of further comprises creating a communication control block for a connection including said packet, wherein sending said data to said destination includes guiding said data by said communication control block (see column 5, line 5 to column 6, line 61).

Claim 5 has been amended to depend from claim 2 rather than claim 1, and thus is nonobvious over the cited references for all the reasons listed regarding claim 2 as well as

claim 1. In addition, Bilansky does not guide, using a communication control block, the data from a packet that had contained both data and headers.

Regarding dependent claim 6, the Office Action states:

> As per claim 6, Bilansky teaches the method of above claims wherein method of further comprises transmitting a second message packet from said host to said network by referencing said communication control block (see column 6, line 5 to column 8, line 63).

Dependent claim 6 has been amended to define transmitting a second message packet containing data and headers from said host to said network by referencing said communication control block. Bilansky, as noted above, transmits either data packets or header packets, unlike amended claim 6.

Independent claim 29:

Regarding independent claim 29, the Office Action states:

> As per claim 29, Bilansky discloses a method for network communication by a host computer having a processor, a memory and a sequential stack of protocol layers (see figures 1-4), the method comprising:
>
> receiving by the host from the network a packet including data and a plurality of headers relating to the stack of protocol layers, said data having a destination in said host, categorizing said packet with a hardware logic sequencer (see figures 1-4 client 100, server 200 and protocol layers 110, 130,150,170, 190), including classifying said headers and choosing, based upon said summary, whether to send said packet to said stark of protocol layers or to bypass said stack of protocol layers by sending said data to said destination (see abstract, column 2, lines 25-50, column 3, line 65 to column 4, line 6, column 4, lines 40-67, column 6, lines 5-45, DDM fast-path bypasses the communication service layers).
>
> Bilansky does not explicitly show the process of processing plurality of headers as a group and creating a summary of group of headers. However, Bilansky does show the processing of headers (see column 5, line 24 to column 6, line 66), Radogna discloses a method for processing messages in a communications network similar to that of Bilansky, wherein Radogna discloses the process of processing plurality of headers as a group and creating a summary of group of headers (see column 3, line 12 to column 4, line 50, RHP 46 for parsing headers). Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Bilansky in view of Radogna by including the process of processing plurality of headers as a group and creating a summary of group of headers, because Bilansky

> suggests the processing of headers. One of ordinary skill in the art would have been motivated to modify and use the method disclosed by Bilansky since Bilansky suggests that bypassing the protocol stack can reduce the communication overhead (see column 10, lines 25-35).

Amended claim 29 is allowable for many of the reasons mentioned above explaining why amended claim 1 is not rendered obvious by Bilansky in view of Radogna are also applicable to amended claim 29. In addition, claim 29 defines in part "categorizing said packet with a hardware logic sequencer," which is not taught or suggested in Bilansky. Moreover, applicants submit that one of ordinary skill in the art would not have modified Bilansky to add such a "hardware logic sequencer," since Bilansky's data packets apparently do not require network frame processing when transferred over Bilansky's fast-path.

Claim 29 has also been amended to recite "transferring said data to a destination in said host," unlike the invention of Radogna, which forwards entire network frames to other network nodes. Radogna does not appear to process transport layer headers, and thus could not transfer the data to "a destination in said host." Also, as mentioned above, Radogna does not teach or suggest bypassing the stack of protocol layers for transferring the data to a destination in the host.

As mentioned above, one of ordinary skill in the art would have been discouraged from modifying Bilansky with Radogna to include the process of processing a plurality of headers as a group and creating a summary of group of headers, because the data packets of Bilansky are separate from the control packets of that invention, and adding headers to the data packets and processing the headers to create a summary would have been thought to slow the fast-path of Bilansky.

Assuming for the sake of argument that one of ordinary skill in the art would have modified Bilansky with Radogna to include the process of processing a plurality of headers as a group and creating a summary of group of headers, the resulting combination would not be applicants invention of claim 29. Claim 29 recites "choosing, dependent upon said summary, whether to process said packet with said stack of protocol layers or to bypass said stack of protocol layers by transferring said data to a destination in said host." Neither Bilansky nor Radogna teaches or suggests making such a choice based on such a summary. Instead, Bilansky chooses a fast-path or slow-path based on

preselected conditions, such as whether the information to be transmitted is control information or data, and Radogna does not teach or suggest any choice.

Dependent Claims 30-34:

Regarding claim 30, the Office Action states:

> As per claim 30, Bilansky discloses a method for network communication by a host computer, wherein said packet is a part of a message having a plurality of packets, and further comprising: receiving by said host from said network a second packet of said message, said second packet including additional data and additional headers, categorizing said second packet with said hardware logic sequencer, including class said additional headers and creating a second packet summary, choosing, based upon said second packet summary, whether to send said second packet to said stack of protocol layers or to bypass said stack of protocol layers and send said additional data to said destination, whereby only one of said first and second packets is sent to said stack of protocol layers (see abstract, column 2, lines 25-50, column 3, line 65 to column 4, line 6, column 4, lines 40-67, column 6, lines 5-45, DDM fast-path bypasses the communication service layers).

Dependent claim 30 has also been amended to recite, in part, choosing whether to send a second packet to the stack of protocol layers or to bypass the stack of protocol layers and send the additional data to the destination, dependent upon a second packet summary, wherein only one of the first and second packets is sent to the stack of protocol layers.

In addition to all the nonobvious differences mentioned above regarding independent claim 29, Bilansky does not teach or suggest a different processing path for different received packets of the same message, dependent upon the summaries of the different packets.

Regarding dependent claim 31, the Office Action states:

> As per claim 31, Bilansky discloses the method of claim 29, further comprising: sending said packet to said stack of protocol layers, processing said packet with said stack of protocol layers and thereby creating a context for said message receiving by said host from said network a related packet including additional data and additional headers, and employing said context for sending said related packet to said destination (see abstract, column 2, lines 25-50, column 3, line 65 to column 4, line 6, column 4, lines 40-67, column 6, lines 5-45, DDM fast-

path bypasses the communication service layers, and slow path through the layers).

Dependent claim 31 has also been amended to recite, in part, sending the packet to the stack of protocol layers, processing the packet with the stack of protocol layers and thereby creating a context including the destination for the message, receiving by the host from the network a related packet including additional data and additional headers, and employing the context for sending the additional data to the destination.

In addition to all the nonobvious differences mentioned above regarding independent claim 29, Bilansky does not teach or suggest processing a packet including data to create a context including the destination in the host. Radogna also does not teach or suggest processing a packet including data to create a context including the destination in the host, as Radogna is directed to forwarding network frames.

Regarding dependent claim 32, the Office Action states:

> As per claim 32, Bilansky discloses method of claim 29, further comprising creating a context for a message including said packet said context defining a connection between said host and a remote host, wherein choosing whether to send said packet to said stack of protocol layers or to bypass said stack of protocol layers includes comparing said context (see abstract, column 2, lines 25-50, column 3, line 65 to column 4 line 6, column 4 lines 40-67, column 6, lines 5-45 column 8, lines 47-65).

Dependent claim 32 has also been amended to recite, in part, creating a context for a message including the packet, the context defining a connection between the host and a remote host, wherein choosing whether to process the packet with the stack of protocol layers or to bypass the stack of protocol layers includes comparing the summary with the context. Bilansky, as noted above, does not create a summary and so matching such a summary with a context is not possible. Even if such a summary should be created by Bilansky, the determination whether to send on a fast or slow path in that invention is based on other criteria, and would not include comparing the summary with the context.

Regarding dependent claim 33, the Office Action states:

> As per claim 33, Bilansky discloses the method of claim 29, further comprising bypassing said stack of protocol layers by sending said data to said destination in a form suitable for said destination (see abstract,

column 2, lines 25-50, column 3, line 65 to column 4, line 6, column 4, lines 40-67, column 6, lines 5-45 DDM fast-path bypasses the communication service layers, and slow path through the layers).

Dependent claim 33 has also been amended to recite, in part, the method of claim 29, further comprising bypassing said stack of protocol layers by sending said data without said headers to said destination in a form suitable for said destination. Bilansky, as noted above, does not have packets containing data and headers, and so sending the data without the headers from those packets is not possible. Even if such a combined data and header packet were used in Bilansky, that reference does not teach or suggest how to split the data and headers of such a packet, or how to land the data without the headers in a destination in a form suitable for the destination.

Regarding claim 34, the Office Action states:

> As per claim 34 Bilansky discloses the method of claim 29, further comprising sending said packet to said stack of protocol layers, processing said packet with said stack of protocol layers and thereby creating a context for said message, and employing said context for transmitting a reply to said -network from said application space, including prepending a transmission header to reply data, said transmission header including control information regarding each of said protocol layers (see abstract, column 2, lines 25-50, column 3, line 65 to column 4, line 6, column 4, lines 40-67, column 6, lines 5-45, DDM fast-path bypasses the communication service layers, and slow path through the layers; see column 8, lines 5-65).

Dependent claim 34 has also been amended to recite, in part, creating a context for a message including the packet, the context defining a connection between the host and a remote host, and employing the context for transmitting a reply to the network from the host, including prepending a transmission header to reply data, the transmission header including control information regarding each of the protocol layers. Bilansky, as noted above, has separate header packets and data packets, and so does not prepend a header to a data packet, since such a header is not only unnecessary for the data packet, but would also slow down if not destroy the fast-path of that invention.

B. The New Claims:

New claim 54 recites the method of claim 1, wherein said destination is a file cache in said host. Support for this claim can be found, for example, on page 25, line 4 of the original specification.

New claim 55 recites the method of claim 1, wherein the host is connected to the network with a network interface device, and said receiving occurs in said device. Support for this claim can be found, for example, on page 25, lines 6-23 of the original specification.

New claim 56 recites the method of claim 1, wherein said summary includes information regarding a transport layer header of said headers. Support for this claim can be found, for example, on page 20, line 20 - page 21, line 14 of the original specification.

New claim 57 recites the method of claim 4, further comprising receiving by said host from said network a second message packet, and transferring said second message packet to said destination by referencing said communication control block. Support for this claim can be found, for example, on page 23, lines 17-18 of the original specification.

New claim 58 recites the method of claim 29, wherein said destination is a file cache in said host. Support for this claim can be found, for example, on page 25, line 4 of the original specification.

New claim 59 recites the method of claim 29, wherein the host is connected to the network with a network interface device, and said receiving occurs in said device. Support for this claim can be found, for example, on page 25, lines 6-23 of the original specification.

New claim 60 recites a method for communication between a network and a host computer having a processor and a stack of protocol layers, the method comprising: a step for receiving, by said host from said network, a message packet including data and a plurality of headers corresponding to said stack of protocol layers, wherein said data has been sent to the host for placement in the host according to protocol processing of said headers, and said headers are made of a series of bytes, a step for categorizing said series of bytes to obtain a status of said packet, and a step for choosing whether to process said packet by said protocol layers, said step for choosing dependent on said status. Support

for this claim can be found, for example, on page 18, line 20 - page 25, line 5 of the original specification.

New claim 61 recites the method of claim 60, further comprising transferring said data to a destination in said host without processing said packet by said protocol layers. Support for this claim can be found, for example, on page 24, line 20 - page 25, line 5 of the original specification.

New claim 62 recites the method of claim 60, wherein said categorizing said series of bytes includes processing a transport-layer header of said plurality of headers. Support for this claim can be found, for example, on page 20, line 20 - page 21, line 14 of the original specification.

New claim 63 recites the method of claim 60, wherein the host is connected to the network with a network interface device, and said receiving occurs in said device. Support for this claim can be found, for example, on page 25, lines 6-23 of the original specification.

New claim 64 recites the method of claim 60, further comprising transferring said data to a destination in said host according to said status. Support for this claim can be found, for example, on page 25, lines 1-5 of the original specification.

C. Other Cited References:

The Office Action also cites two other references considered pertinent to the present application:

(a) U.S. Patent Number 5,930,830 to Mendelson et al., entitled "System and Method for Concatenating Discontiguous Memory Pages," and

(b) U.S. Patent Number 6,061,368 to Hitzelberger, entitled "Custom Circuitry for Adaptive Hardware Routing Engine."

Mendelson et al. involves receiving blocks of data that may be discontiguous or exceed the size of a memory page for a processor and concatenating that data.

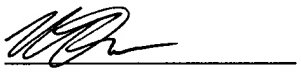
Hitzelberger involves circuitry for routing including a staggered multiplexer for implementing a hashing algorithm.

Neither of these references teaches or suggests the claims at issue.

Conclusion

Applicants would like to thank the Examiner for his work and attention to this case. Applicants believe that the application, as amended, is in condition for allowance, and a Notice of Allowance is solicited.

Respectfully submitted,



Mark Lauer
Reg. No. 36,578
7041 Koll Center Plaza,
Suite 280
Pleasanton, CA 94566
Tel: (925) 484-9295
Fax: (925) 484-9291

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231, on August 17, 2000.

Date: 8-17-2000 

Mark Lauer